

WHAT IS CLAIMED IS:

1. A data processor for communicating data to and from a host computer via an interface with a packet including a packet ID, the data processor comprising:
5 a packet recognition circuit for receiving the packet including the packet ID from the host computer and recognizing the type of the packet from the packet ID; and
10 a packet length measuring circuit connected to the packet recognition circuit for measuring packet length of the packet received from the host computer and determining whether the measured packet length is in accordance with the packet type recognized by the packet determination circuit.

15 2. The data processor according to claim 1, wherein the packet length measuring circuit generates a suspension signal for suspending data transfer from the data processor to the host computer when the measured packet length is not in accordance with the packet type.

20 3. The data processor according to claim 1, wherein the interface includes a USB interface.

4. A data processor for communicating data to and from a host computer via an interface, the data processor comprising:
25 a plurality of end points, each processing a transaction corresponding to a data transfer request from the host computer, wherein each of the end points stores a data toggle bit having a value that is inverted whenever a predetermined packet from the host computer is received; and
30 a toggle bit switching circuit connected to the end points to determine whether a first end point that received

a transfer request in a previous transaction and a second end point that received a transfer request in the present transaction are substantially the same, wherein the toggle bit switching circuit inverts the value of the data toggle bit stored in the first end point when the first end point differs from the second end point.

5 5. The data processor according to claim 4, wherein:
 each end point has an end point number, and the
10 predetermined packet includes the end point number; and
 the toggle bit switching circuit includes an end point
 storage section for storing the end point number included in
 the predetermined packet transmitted from the host computer
 and determines whether the first and second end points are
15 substantially the same by comparing the end point number of
 the first end point stored in the storage section with the
 end point number of the second end point.

20 6. The data processor according to claim 5, wherein
 the interface includes a USB interface, and the end point is
 included in a token packet.

25 7. The data processor according to claim 4, wherein
 the toggle bit switching circuit inverts the data toggle bit
 value of the first end point when receiving a handshake
 packet from the host computer.

30 8. The data processor according to claim 4, wherein
 the interface includes a USB interface.

 9. A data processor for communicating data to and
 from a host computer via a predetermined interface with a
 packet including a packet ID, the data processor comprising:

a packet recognition circuit for receiving the packet including the packet ID from the host computer and recognizing the type of the packet from the packet ID;

5 a packet length measuring circuit connected to the packet recognition circuit for measuring packet length of the packet received from the host computer and determining whether the measured packet length is in accordance with the packet type recognized by the packet determination circuit;

10 a plurality of end points, each processing a transaction corresponding to a data transfer request from the host computer, wherein each of the end points stores a data toggle bit having a value that is inverted whenever receiving a predetermined packet from the host computer; and

15 a toggle bit switching circuit connected to the end points to determine whether a first end point that received a transfer request in a previous transaction and a second end point that received a transfer request in the present transaction are substantially the same, wherein the toggle bit switching circuit inverts the value of the data toggle
20 bit stored in the first end point when the first end point differs from the second end point.

10. The data processor according to claim 9, wherein the interface includes a USB interface.

25

11. A method for recognizing the type of a packet, which includes a predetermined packet length and a packet ID, in a data processor communicating data to and from a host computer via an interface, the method comprising:

30 receiving the packet from the host computer via the interface;

recognizing the type of the received packet from the packet ID;

measuring a packet length of the received packet; and
determining whether the measured packet length is in
accordance with the packet type recognized from the packet
ID.

5

12. The method according to claim 11, further
comprising:

suspending transfer of data related with the received
packet when the measured packet length is not in accordance
10 with the packet type recognized from the packet ID.

13. The method according to claim 11, wherein the
interface includes a USB interface.

15 14. A method for correcting an error in a data
processor communicating data to and from a host computer via
an interface, wherein the data processor includes a
plurality of end points, each processing a transaction
corresponding to a data transfer request from the host
20 computer, and each of the end points storing a data toggle
bit having a value that is inverted whenever a normal
receipt acknowledgement from the host computer is received,
and the error is a data toggle bit error in which the data
toggle bit is not inverted when the data processor does not
25 receive the normal receipt acknowledgement from the host
computer, the method comprising:

determining whether a first end point of the plurality
of end points that received a transfer request in a previous
transaction and a second end point of the plurality of end
30 points that received a transfer request in the present
transaction are substantially the same; and

inverting the value of the data toggle bit stored in
the first end point when the first end point differs from

the second end point.

15. The method according to claim 14, further comprising:

5 re-transmitting data in accordance with the value of the data toggle bit stored in the first end point when the first and second end points are substantially the same.

10 16. The method according to claim 14, wherein each of the end points has an end point number, and said determining includes determining whether the first and second end points are substantially the same by comparing the end point number of the first end point with the end point number of the second end point.

15

17. The method according to claim 14, wherein the interface includes a USB interface.

18. A method for controlling a data processor
20 communicating data to and from a host computer via a USB interface, wherein the data processor includes a plurality of end points, each processing a transaction corresponding to a data transfer request from the host computer, and each of the end points storing a data toggle bit having a value
25 that is inverted whenever a handshake packet from the host computer is received, the method comprising:

determining whether a first end point of the plurality of end points that received a transfer request in a previous transaction and a second end point of the plurality of end
30 points that received a transfer request in the present transaction are substantially the same; and

inverting the value of the data toggle bit stored in the first end point when the first end point differs from

the second end point.

19. The method according to claim 18, further comprising:

5 re-transmitting data in accordance with the value of the data toggle bit stored in the first end point when the first and second end points are substantially the same.

10 20. The method according to claim 18, wherein each of the end points has an end point number, and said determining includes determining whether the first and second end points are substantially the same by comparing the end point number of the first end point with the end point number of the second end point.